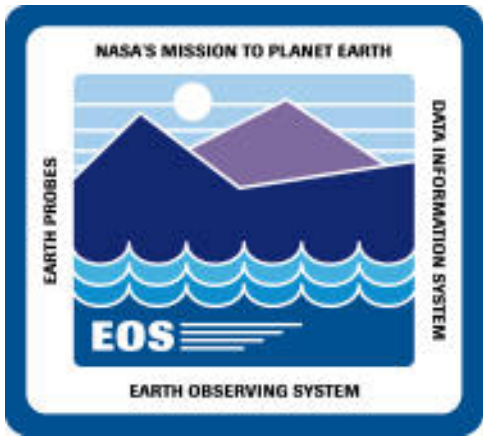


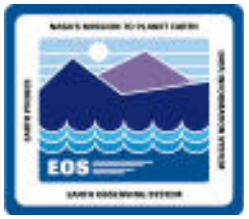

# Population of Granule Metadata

## Richard Morris

**rmorris@eos.hitc.com**

**19 April 1996**

# HDF-EOS Metadata Context



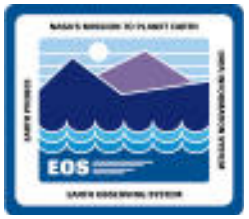

**Different categories of products managed by ECS require different levels of Metadata.**

- **Full level of Metadata - required for products generated within EOSDIS i.e., EOS-HDF files produced by PGE**
- **Intermediate level of Metadata - required for products generated outside EOSDIS but ingested and used within EOSDIS**
- **Limited level of Metadata - applies to all other data sets. Very few are expected to fall into this category.**

**<u>FOCUS</u>**

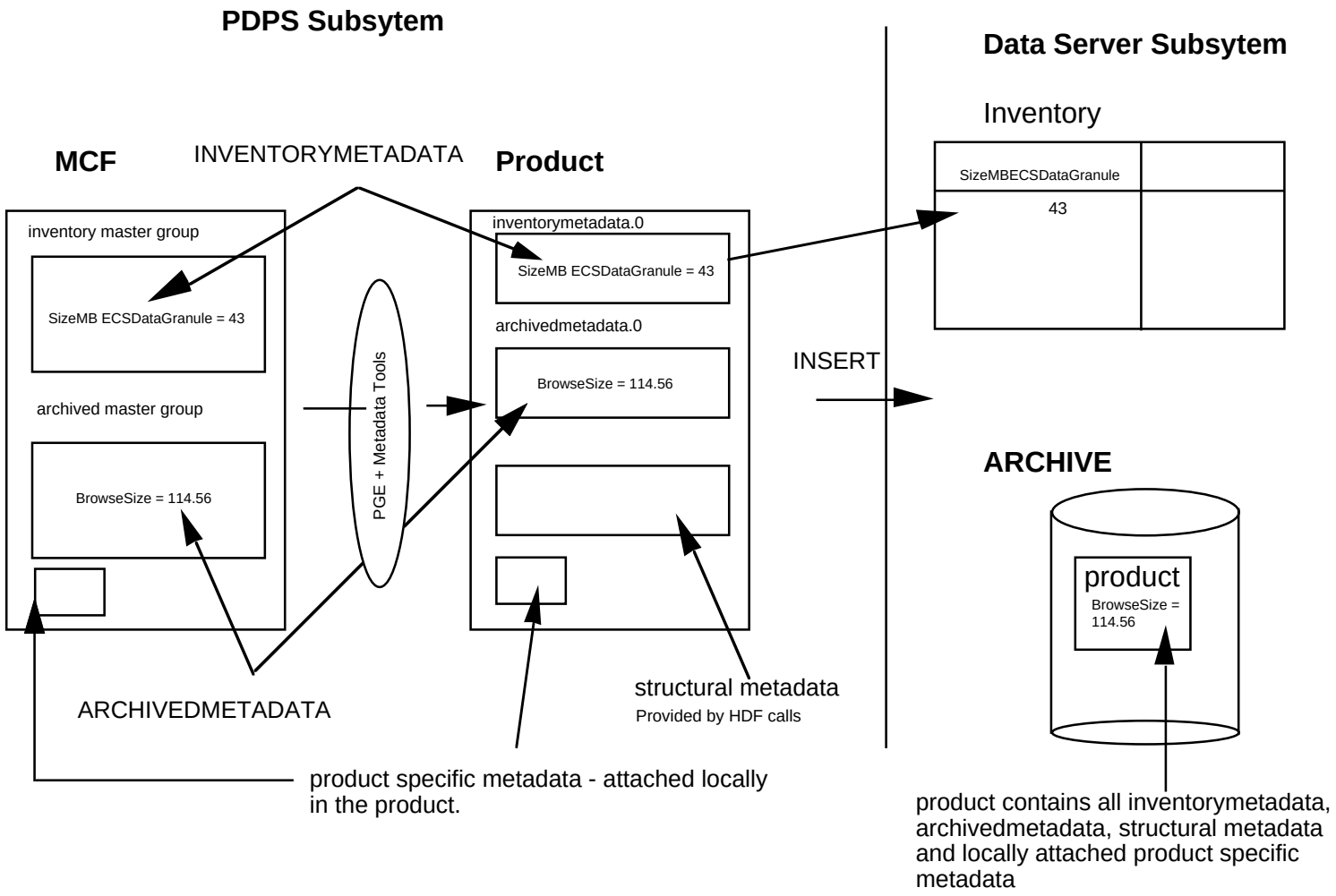
- **This presentation will concentrate on the tools for providing Metadata to the product in HDF-EOS format. These tools will be utilized by the PGEs.**

# ECS Metadata

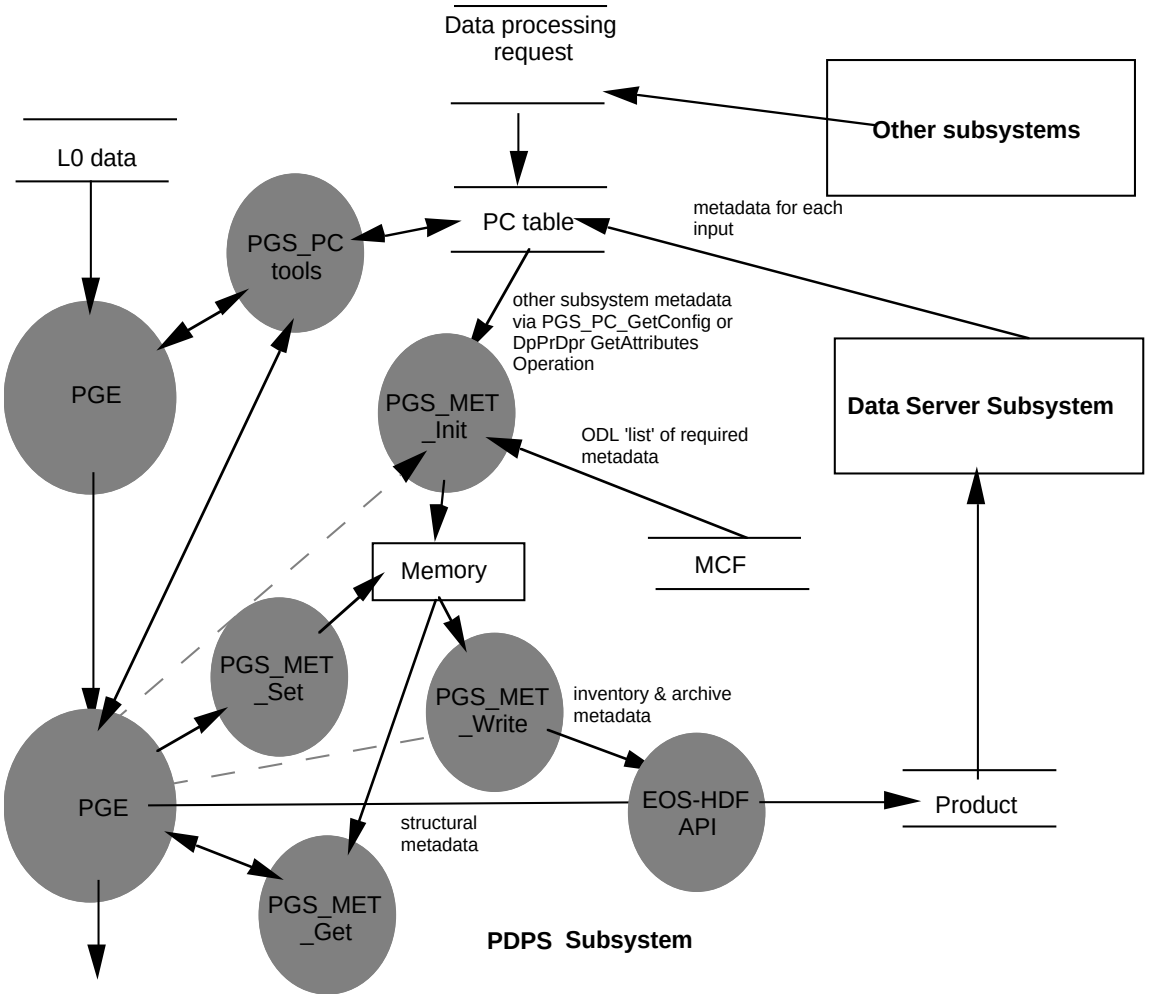



- **Metadata in each granule**
  - **Inventory**
  - **Archived**
  - **Structural**
  - **Product Specific**
- **Inventory metadata duplicated and used to populate the inventory.**
- **All inventory and archived metadata stored in EOS products.**
- **Inventory, Archived and Product Specific are attached to the granule using Metadata Toolkit calls.**
- **Structural metadata describes the HDF structure and some of this will be used for performing “services” on the data.**
- **Structural Metadata is created and attached to the granule by the HDF-EOS library routines.**

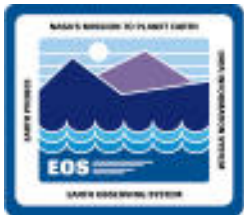
# Metadata Relationships



# Metadata - Data Producer View

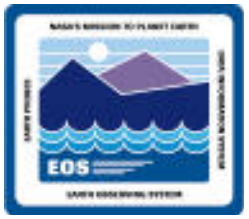


# Metadata Configuration File MCF




- MCF is the means by which the Metadata tools attach values to attributes which describe the HDF-EOS file.
- MCF is a template which can be customized by the PGE developer.
- Allows metadata values to be attached to an attribute
  - values can come from PGE, MCF or PCF
- MCF is in ODL/PeV format (Object Description Language, Parameter equals Value).
- Once MCF in memory is filled using the Metadata tools, it is attached to the HDF product.
  - attached as a global attribute - inventorymetadata.0, archivedmetadata.0
  - attached as a local attribute - local hdf attribute name.

# MCF Example



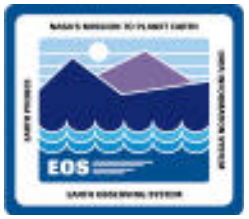

**EXAMPLE of ODL/PeV structure from an MCF**

```
GROUP = CircleGroup
    OBJECT = CenterLatitude
    Data_Location = “PGE”
    TYPE = “DOUBLE”
    NUM_VAL = 1
    Mandatory = “TRUE”
    END_OBJECT = CenterLatitude
END_GROUP = CircleGroup
```

# SDP Toolkit Metadata API Table

| Category | Routine Name - C and FORTRAN | Description |
|---|---|---|
| Initialise | PGS_MET_Init | Initialise the MCF into memory, Some initial checking performed, values located in the Process Control File (PCF) retrieved |
| Set | PGS_MET_SetAttr | Set the value of an attribute. |
| Write | PGS_MET_Write | Write a number of attributes to an HDF-EOS file header |
| Retrieve | PGS_MET_GetSetAttr | Retrieve attributes which are already set in the MCF |
| Retrieve | PGS_MET_GetConfigData | Retrieve Configuration Data from the PCF |
| Retrieve | PGS_MET_GetPCAttr | Retrieve an attribute value from a file referenced in the PCF - value may be in flat ascii file, or HDF-EOS file written by the Metadata tools |
| Delete | PGS_MET_Remove | Delete the MCF held in memory |

# SDP Toolkit Metadata API




**The following scenario outlines the proposed use of the Metadata toolkit.**

**Full details may be found in the Toolkit Users Guide Appendix J (document 003-CD-003-001 July 1995).**

***STEP 1—Initialize MCF (load MCF customized template into memory)***

- **PGS_MET_Init(filelogical, metadata handles)**

***STEP 2—Extract Value from a file referenced in PCF - value may be in flat ascii file, or an HDF file written by the metadata tools***

- **PGS_MET_GetPCAttr(product file id, product version number, name of hdf attribute containing metadata, metadata parameter, returned metadata parameter value)**

***STEP 3—Write the value extracted to the MCF in memory***

- **PGS_MET_SetAttr(metadata group name, name.class of parameter, value to be inserted)**

# SDP Toolkit Metadata API (cont.)

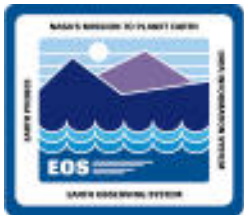
***STEP 4—A value already held in the MCF in memory is needed to calculate a new value for a product specific object.***

- **PGS_MET_GetSetAttr(metadata group name, name.class of parameter, value to be passed back)**

***STEP 5—In order to calculate this new value, information is also needed from the Configuration parameters set up in the Process Control File.***

- **PGS_MET_GetConfigData(name of parameter, value to be passed back)**
- **PCF file entry would look like this:**
- **10255|FOO|”BAR”|/location/file|version**

# SDP Toolkit Metadata API (cont.)




***STEP 6—The PGE has used the two inputs to calculate a new value for one of the MCF objects, and wants to write it to the MCF held in memory.***

- **PGS_MET_SetAttr(metadata group name, name.class of parameter, value to be inserted)**

***STEP 7—After multiple calls to PGS_MET_SetAttr the MCF in memory is now complete, all the granule specific metadata has been set, the relevant product specific metadata has been set, the PGE now writes the metadata out as an HDF attribute attached to the product.***

- **PGS_MET_Write(metadata group to be written out, HDF file attribute name ,HDF file ID)**

***STEP 8 - If the user wants to write out the MCF in memory as an ASCII file this is possible using PGS_MET_Write with a number of different inputs.***

- **PGS_MET_Write(metadata group to be written out, HDF file attribute name [MUST BE SET TO NULL (char *)NULL],HDF file ID [MUST BE SET TO NULL (PGSt_Integer)NULL])**